

1/28

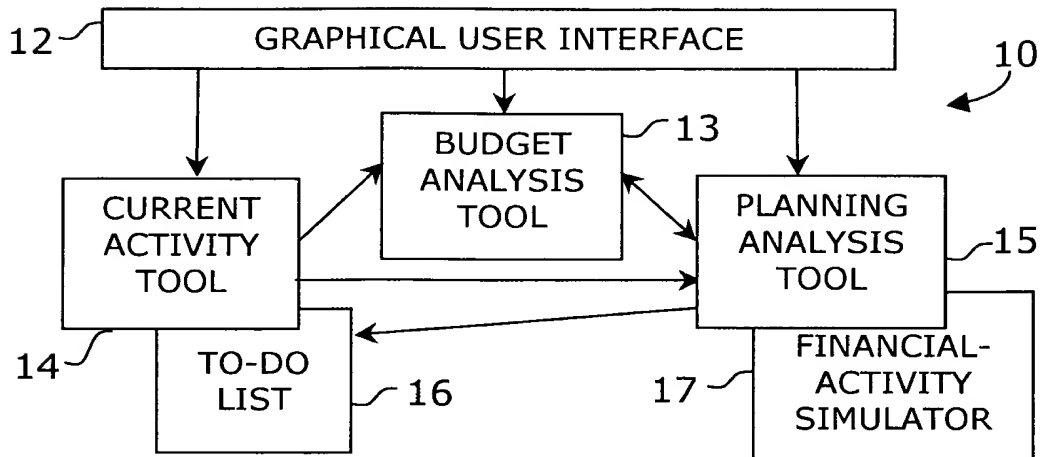


FIG. 1

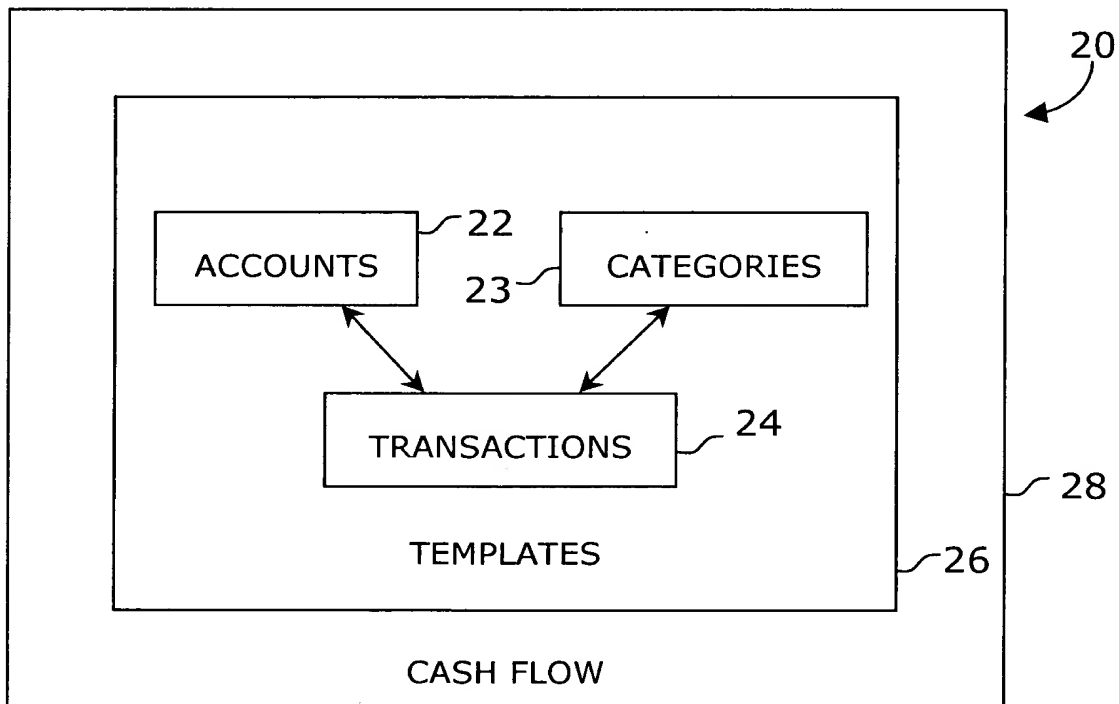


FIG. 2

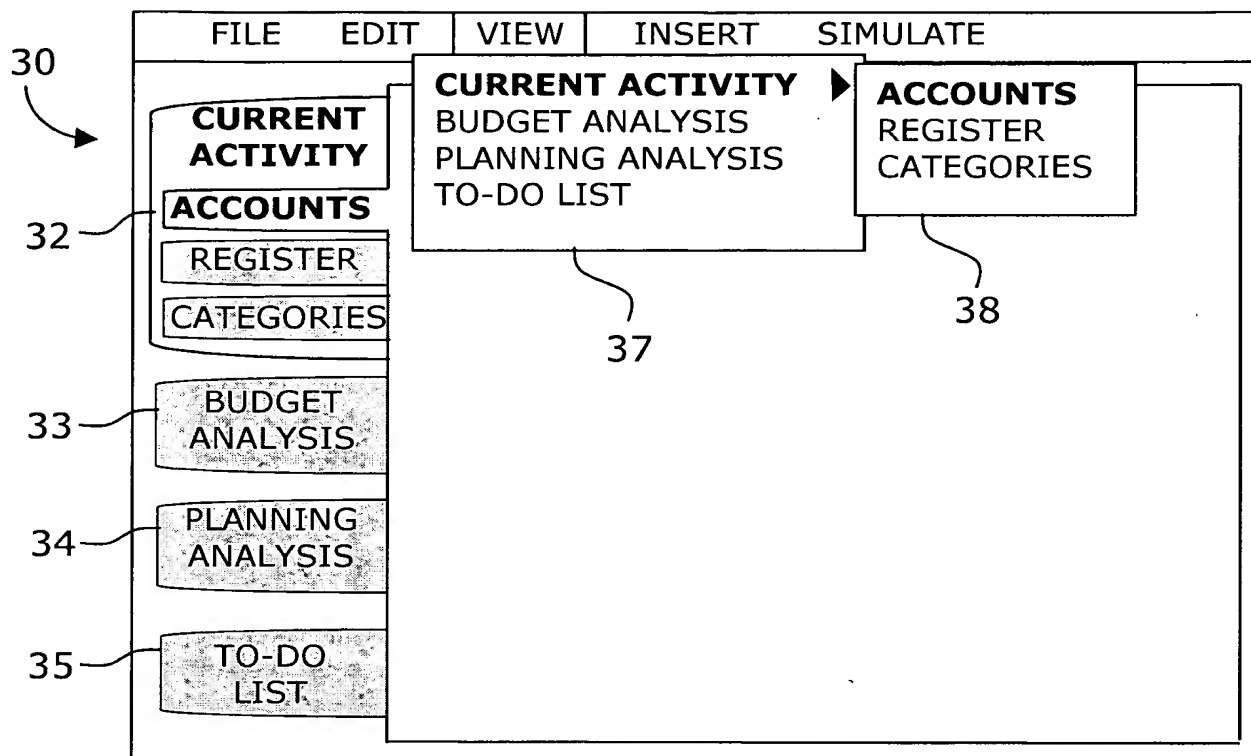


FIG. 3

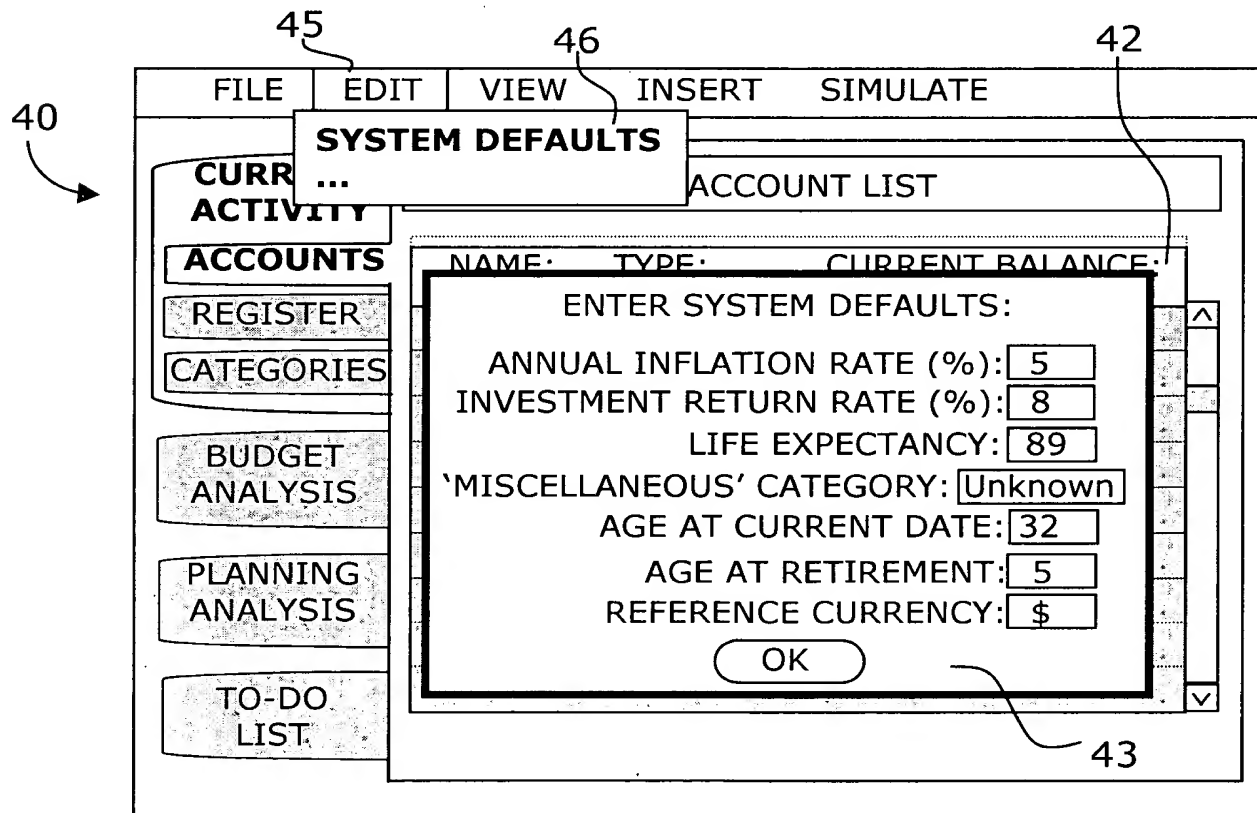


FIG. 4

3/28

54 55

52

50

FILE EDIT VIEW INSERT SIMULATE

**IMPORT**  
...  
**ACTIVITY**  
**ACCOUNTS**  
REGISTER  
CATEGORIES  
BUDGET ANALYSIS  
PLANNING ANALYSIS  
TO-DO LIST

**ACCOUNT**  
CATEGORY  
TEMPLATE  
TRANSACTION  
...

**BANK**  
CREDIT  
LOAN  
MORTGAGE  
...

T BALANCE:

**OPEN/CREATE BANK ACCOUNT:**  
 NAME OF ACCOUNT: MY CHECKING  
 OPENING DATE: 01/01/99  
 INITIAL BALANCE (\$): 5000  
 ANNUAL INTEREST RATE (%): 2  
 INTEREST ACCRUED: MONTHLY  
 ...ON THE: 1<sup>ST</sup> OF MONTH  
 CATEGORY FOR INTEREST: UNKNOWN  
 TAXABLE?: YES  
 OK

FIG. 5

64 65

62

60

FILE EDIT VIEW INSERT SIMULATE

**IMPORT**  
...  
**ACTIVITY**  
**ACCOUNTS**  
REGISTER  
CATEGORIES  
BUDGET ANALYSIS  
PLANNING ANALYSIS  
TO-DO LIST

**ACCOUNT**  
CATEGORY  
TEMPLATE  
TRANSACTION  
...

**EXPENSE**  
INCOME

CURRENT BALANCE:

MY CHECKING BANK (\$) 5000

**CREATE EXPENSE CATEGORY:**  
 NAME OF CATEGORY: CLOTHING  
 TAXABLE?: YES  
 DONE ADD BUDGET/PLAN INFO

FIG. 6

4/28

70

74

75

72

FILE EDIT VIEW INSERT SIMULATE

CURRENT ACTIVITY

ACCOUNT

REGISTER

CA

B

AN

PL

AN

T

ACCOUNT CATEGORY

TEMPLATE

TRANSACTION

...

...

SCHEDULED SPENDING

SCHEDULED INCOME

...

**SPENDING ACTIVITY TEMPLATE:**

SPENDING CATEGORY IS: CLOTHING

DESCRIPTION: PERSONAL SPENDING ON CLOTHING

START DATE: 12/1/98 END DATE: <END OF TIME>

SPEND (\$) 200 ON A MONTHLY BASIS,

FROM ACCOUNT: MY CHECKING

INFLATE AT ANNUAL RATE (%): <INFLATION = 5%>

INFLATE ON YEARLY BASIS, ON 1<sup>ST</sup> OF JANUARY

DONE

FIG. 7

83

82

88

86

80

85

87

89

84

FILE EDIT VIEW INSERT SIMULATE

PLANNING ANALYSIS

ALL

TEMPLATE

OBJECT

BUDGET ANALYSIS

CURRENT ACTIVITY

TO-DO LIST

START OF PLAN

Dec 1998

Feb 1999

Apr 1999

Jun 1999

Aug 1999

CATEGORY: CLOTHING

TRANSACTION: MY CHECKING TO CLOTHING

ACCOUNT(BANK): MY CHECKING

1998

2060

VIEW

FIG. 8



5/28

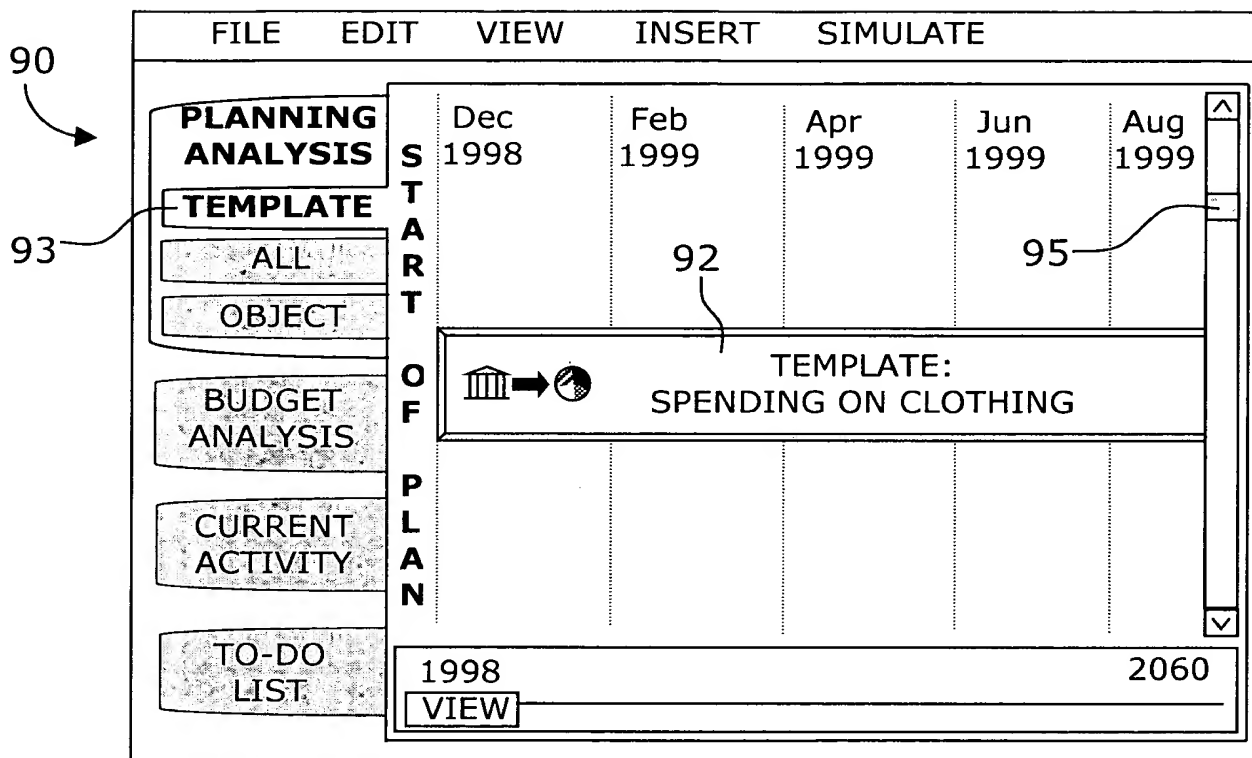


FIG. 9

6/28

FILE

EDIT

VIEW

INSERT

SIMULATE

104

CURRENT ACTIVITY

106

REGISTER

107

ACCOUNTS

CATEGORIES

BUDGET ANALYSIS

PLANNING ANALYSIS

TO-DO LIST

102

MY CHECKING

MY SAVINGS

MY MUTUAL

DATE:

DESCRIPTION:

CATEGORY:

AMOUNT:

CURR. BUDGET STATUS:

5/1/99	BIG SAVINGS	GROCERIES	\$20.82	ON BUDGET
5/2/99	SUPER GAS	AUTO	\$12.43	UNDER BY \$214, .2%, 1 WK
5/3/99	FARLEYS	CLOTHING	\$115.22	UNDER BY \$100, .1%, 2WKS
5/5/99	CITY POWER	ELECTRIC	\$29.43	ON BUDGET
5/7/99	TOYS&MORE	RECREATION	\$54.62	UNDER BY \$19, .1%, 3DAYS
5/9/99	A+ COMPUTERS	COMPUTER	\$82.20	ON BUDGET

105

CURRENT BUDGET INFORMATION FOR CATEGORY: **CLOTHING**

YOU ARE \$100.00 **UNDER** THE PLANNED BUDGET FOR THIS CATEGORY.

YOUR MONTHLY BUDGET FOR THIS CATEGORY IS \$200, SO YOU ARE 2 WEEKS AND .1% BEHIND YOUR BUDGET FOR DATE RANGE 11/98 TO 05/99.

SELECT THE 'BUDGET ANALYSIS' TAB NOW IF YOU WISH TO ALTER THE BUDGET FOR THIS CATEGORY.

100

FIG. 10

103



7/28

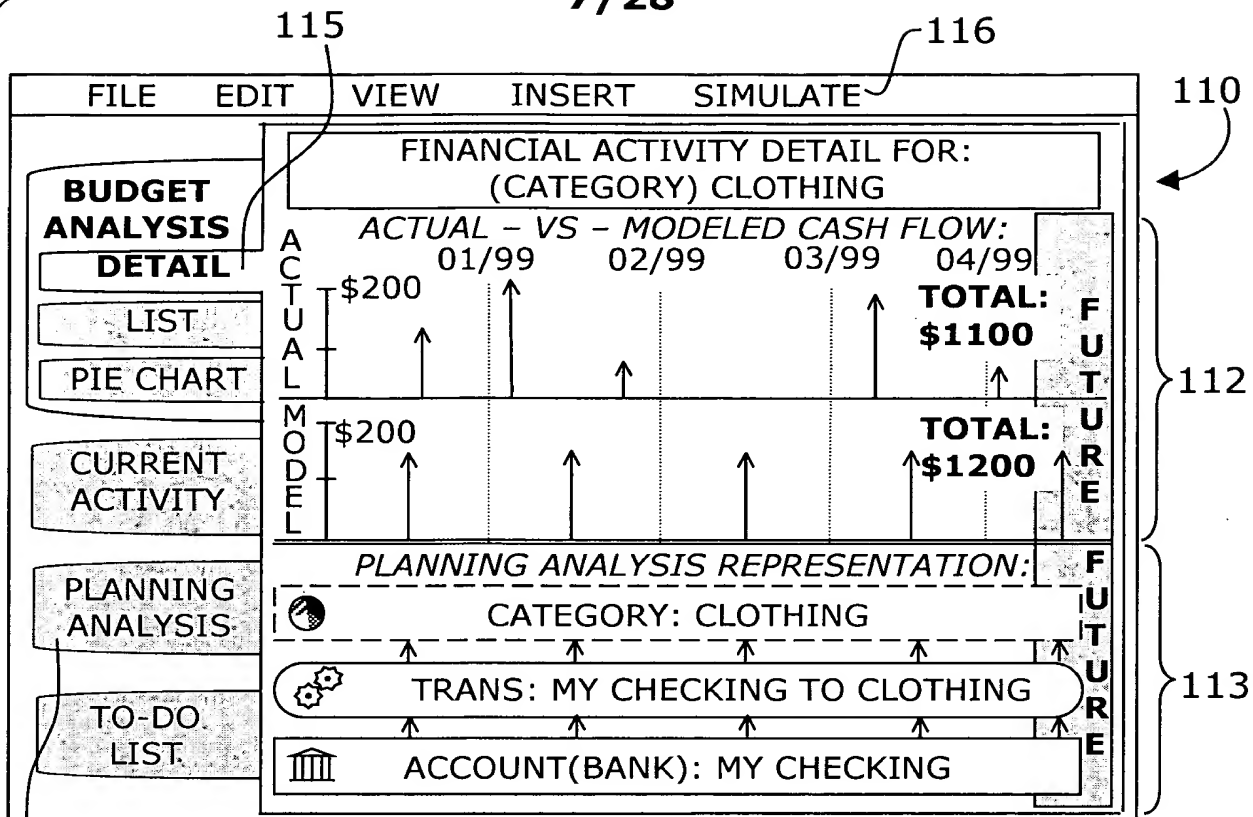


FIG. 11

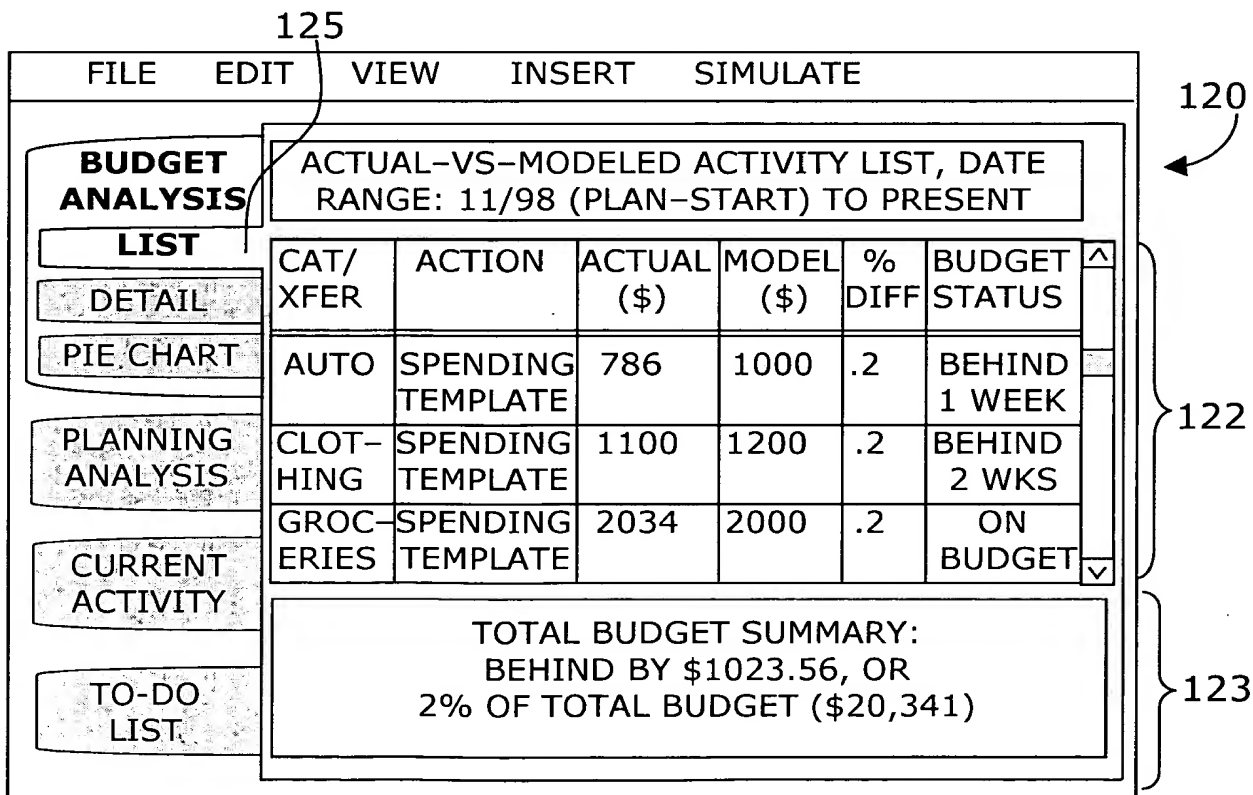


FIG. 12

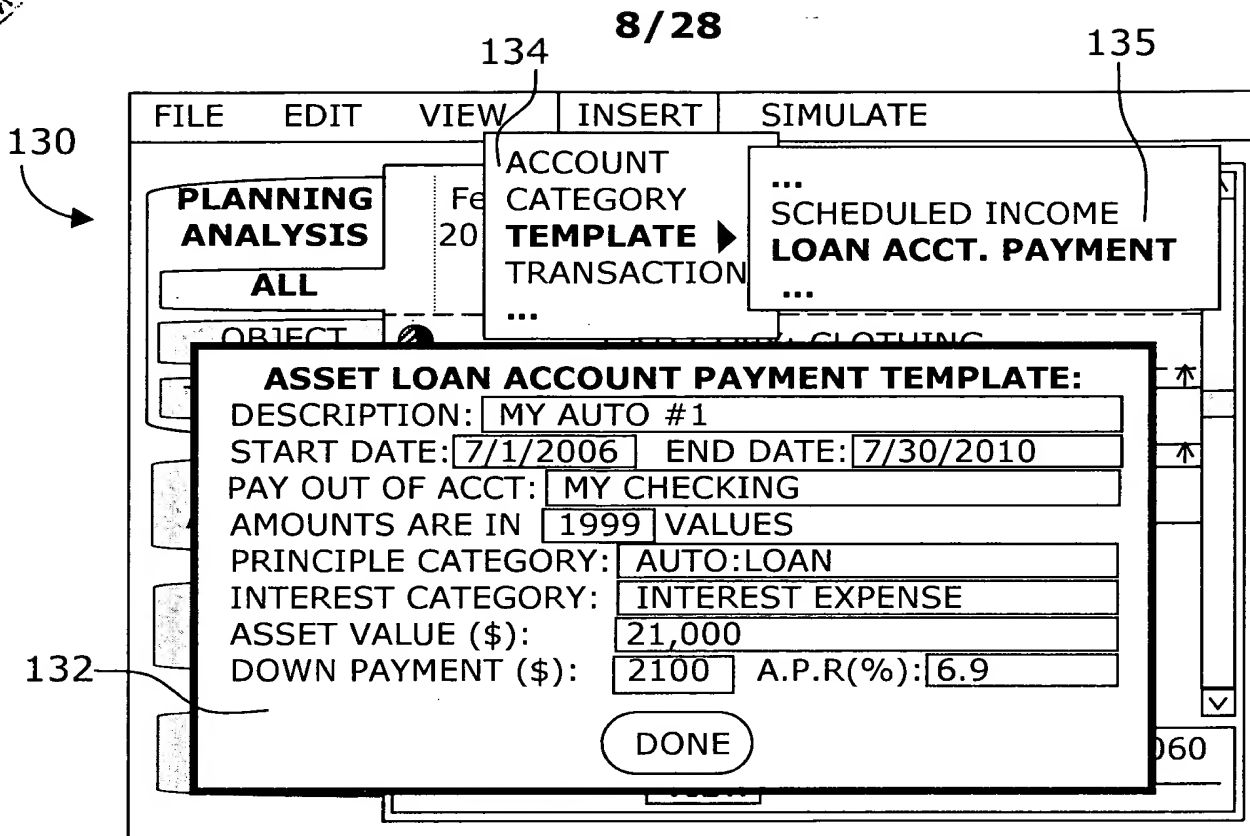


FIG. 13

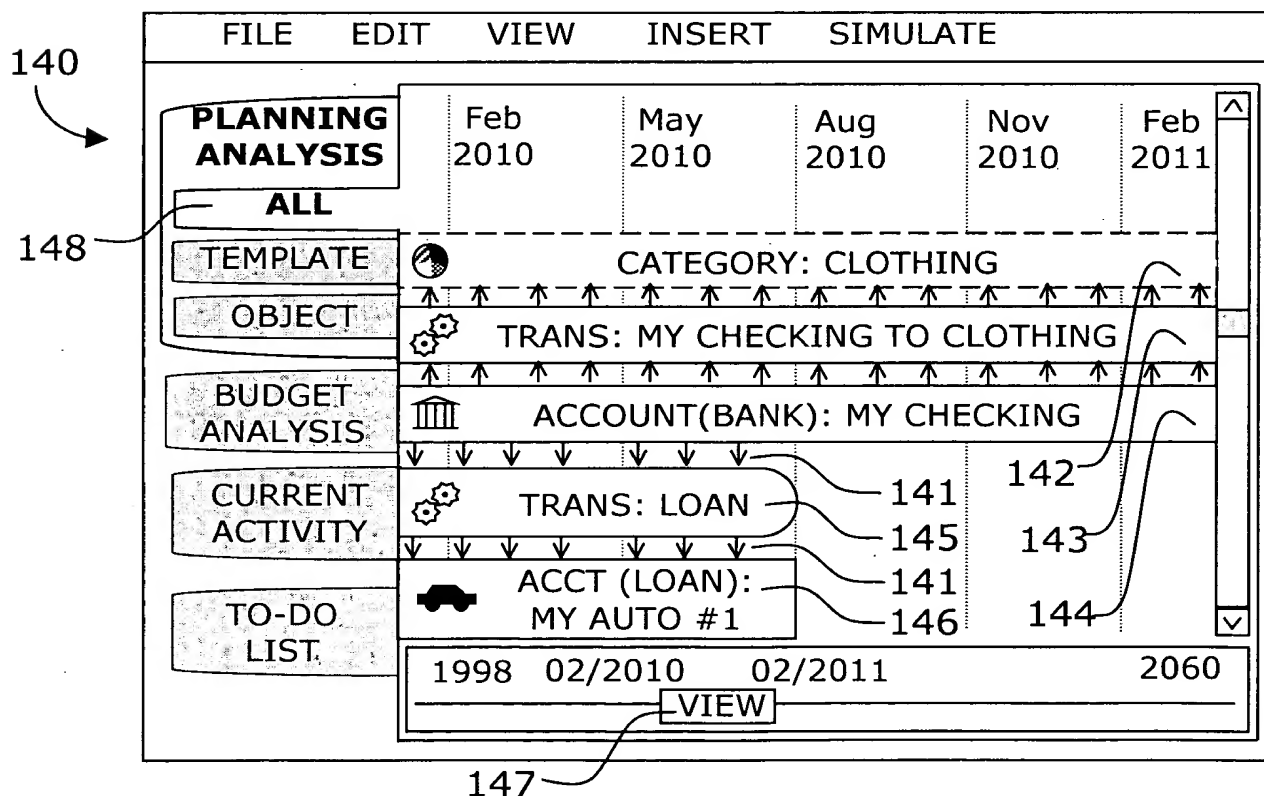


FIG. 14



9/28

150

154

FILE EDIT VIEW INSERT SIMULATE

**PLANNING ANALYSIS TEMPLATE**

ALL

OBJECT

BUDGET ANALYSIS

CURRENT ACTIVITY

TO-DO LIST

Feb 2010 May 2010 Aug 2010 Nov 2010 Feb 2011

TEMPLATE: SPENDING ON CLOTHING

152

TEMPLATE: LOAN MY AUTO #1

153

1998 02/2010 02/2011 2060

VIEW

FIG. 15

160

163

164

FILE EDIT VIEW INSERT SIMULATE

**PLANNING ANALYSIS**

ALL

OBJECT

ACCOUNT CATEGORY TEMPLATE TRANSACTION

... LOAN ACCT. PAYMENT VACATION ACTIVITY ...

**VACATION ACTIVITY TEMPLATE:**

DESCRIPTION: Summer Vacation

START DATE: 5/20/2010 END DATE: 5/25/2010

PAY OUT OF ACCT: MY CHECKING

AMOUNTS ARE IN 1999 VALUES

FOR... SPEND... REPEAT... USE CATEGORY:

AIR FARE \$1200 ONE TIME RECREATION:TRIPS

DINING \$50 DAILY DINING

GIFTS \$50 DAILY GIFTS

DONE

162

060

FIG. 16

10/28

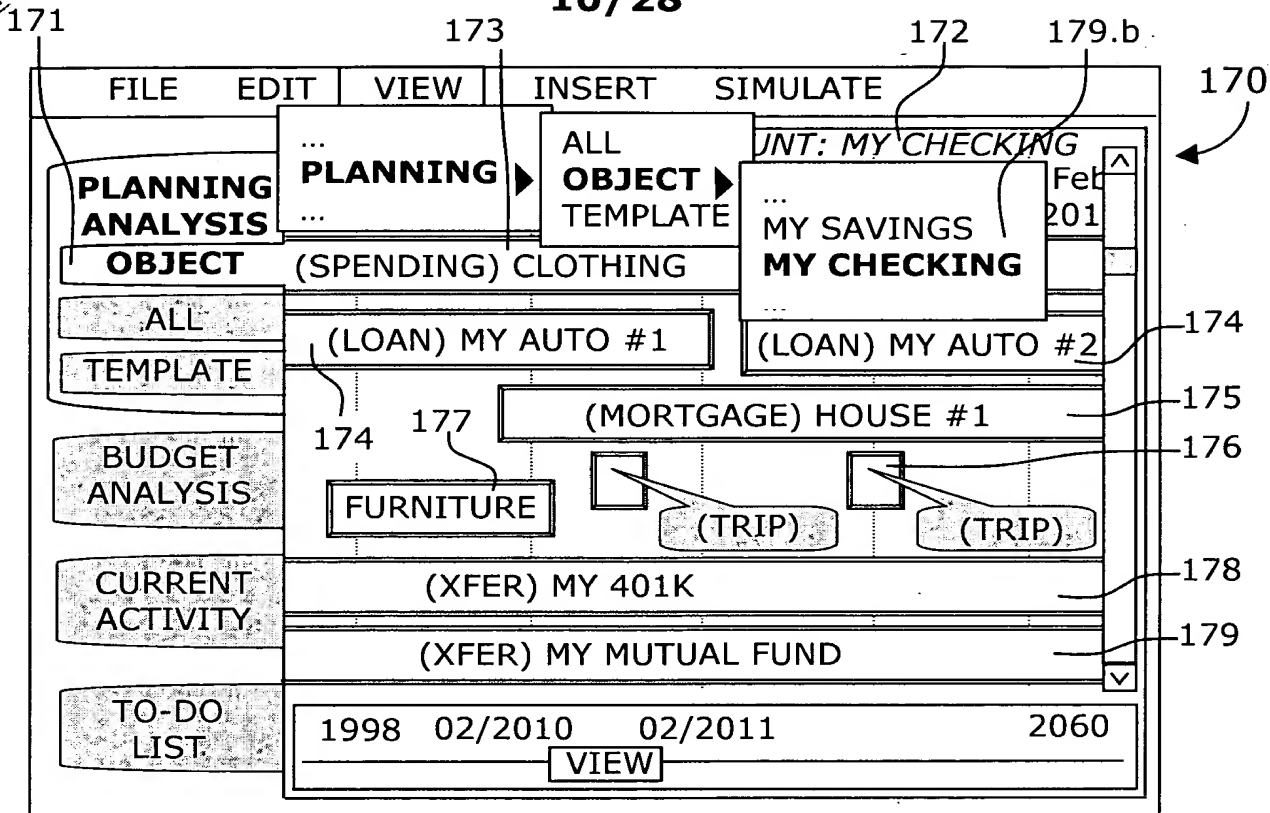


FIG. 17

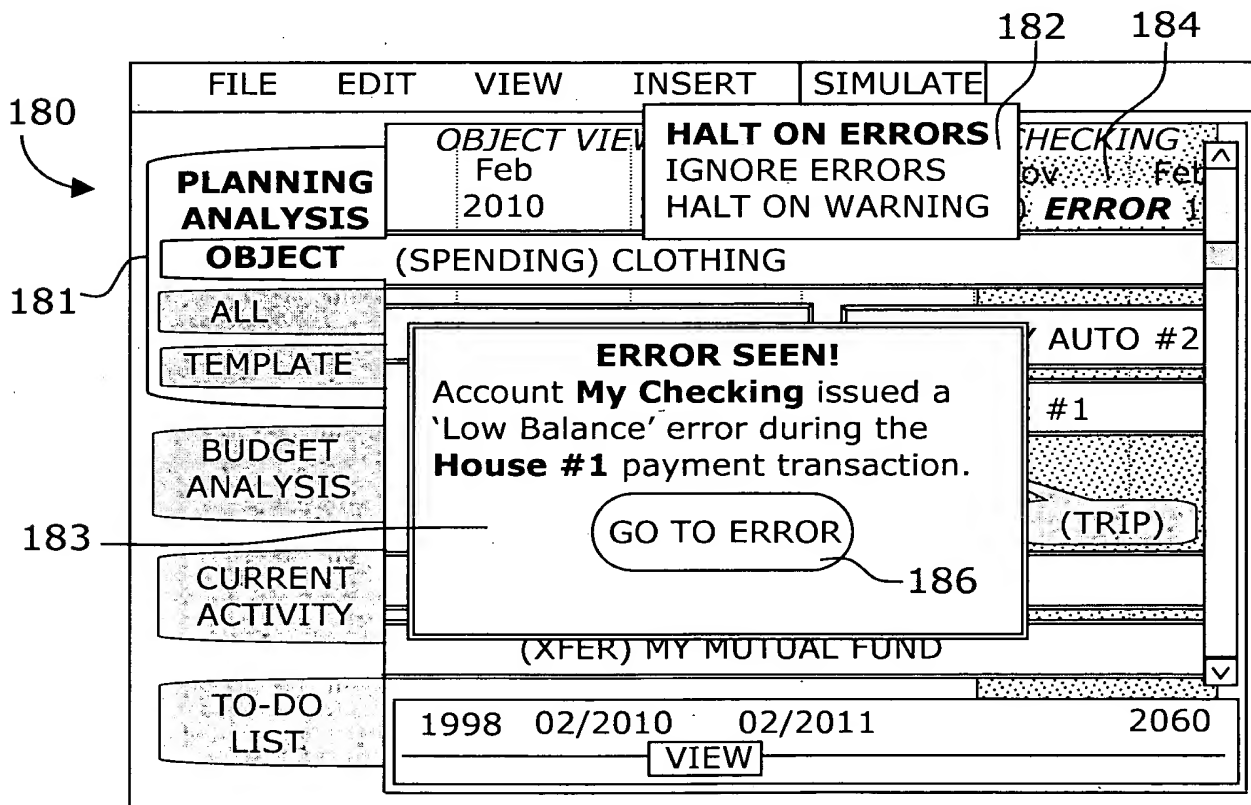


FIG. 18

11/28

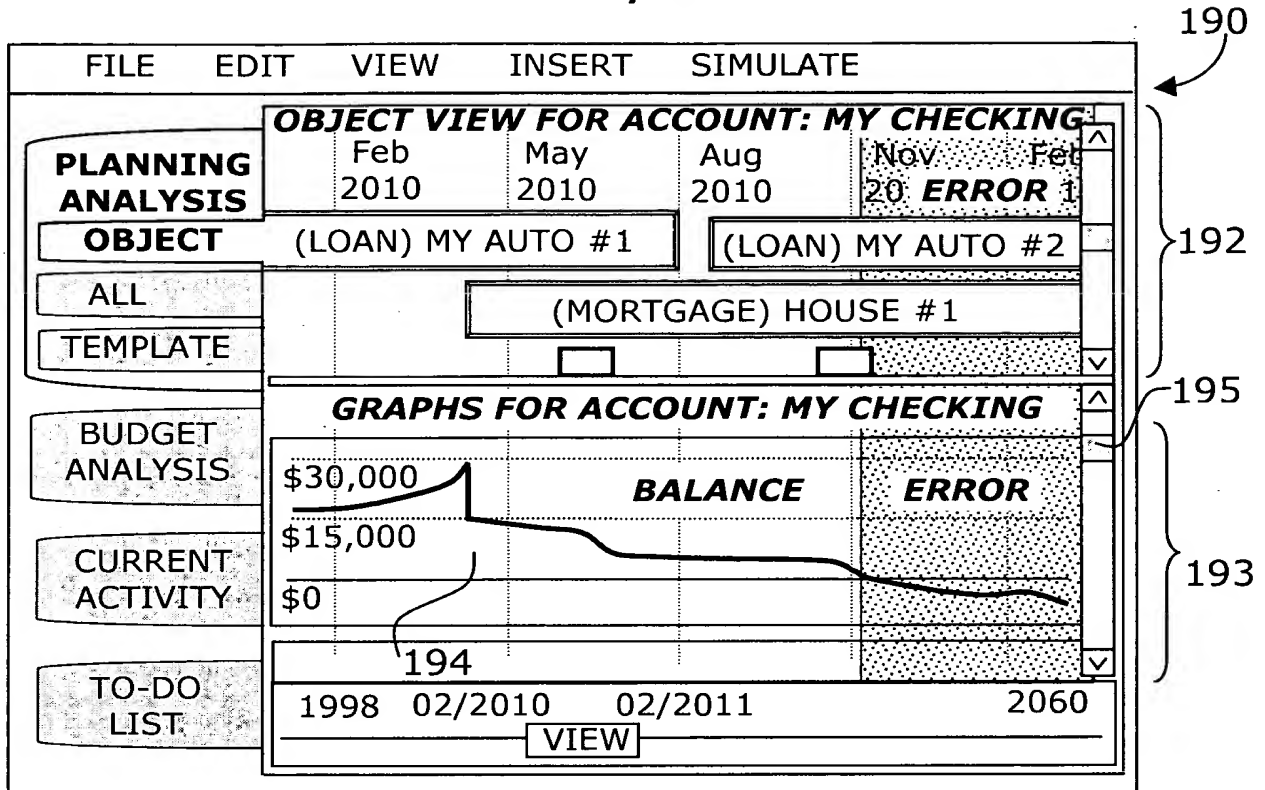


FIG. 19

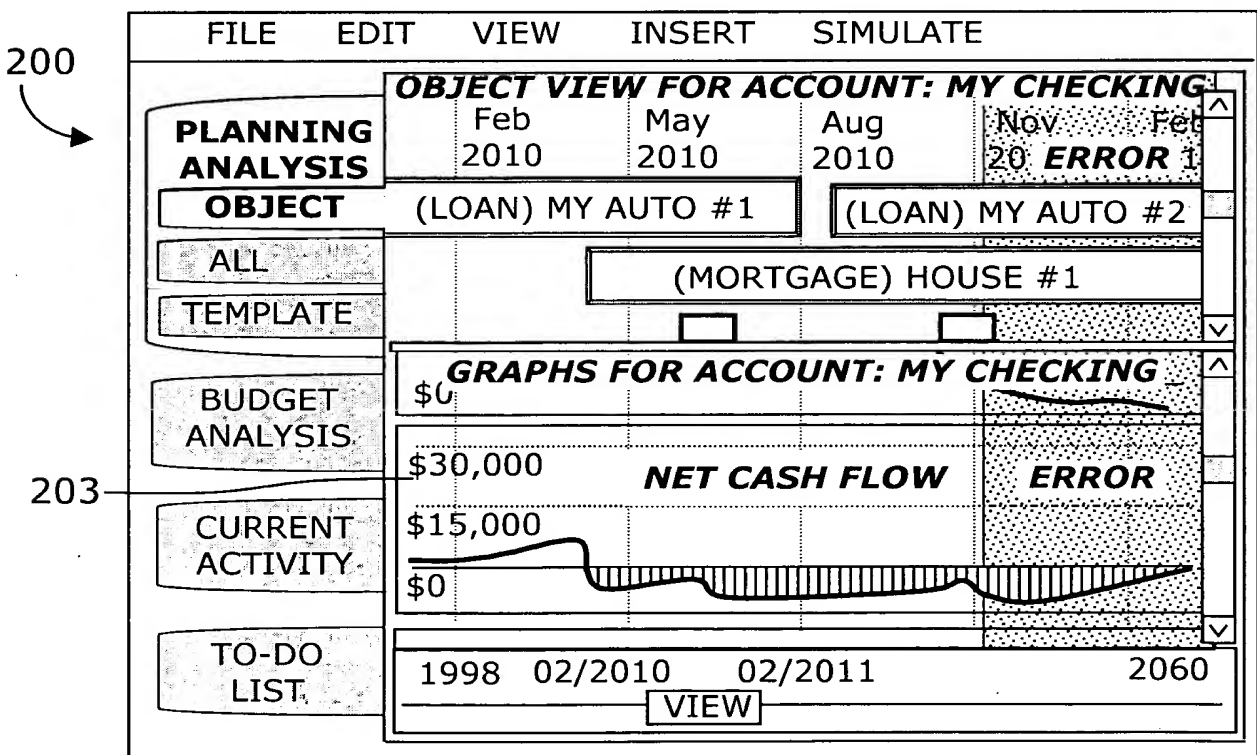


FIG. 20

12/28

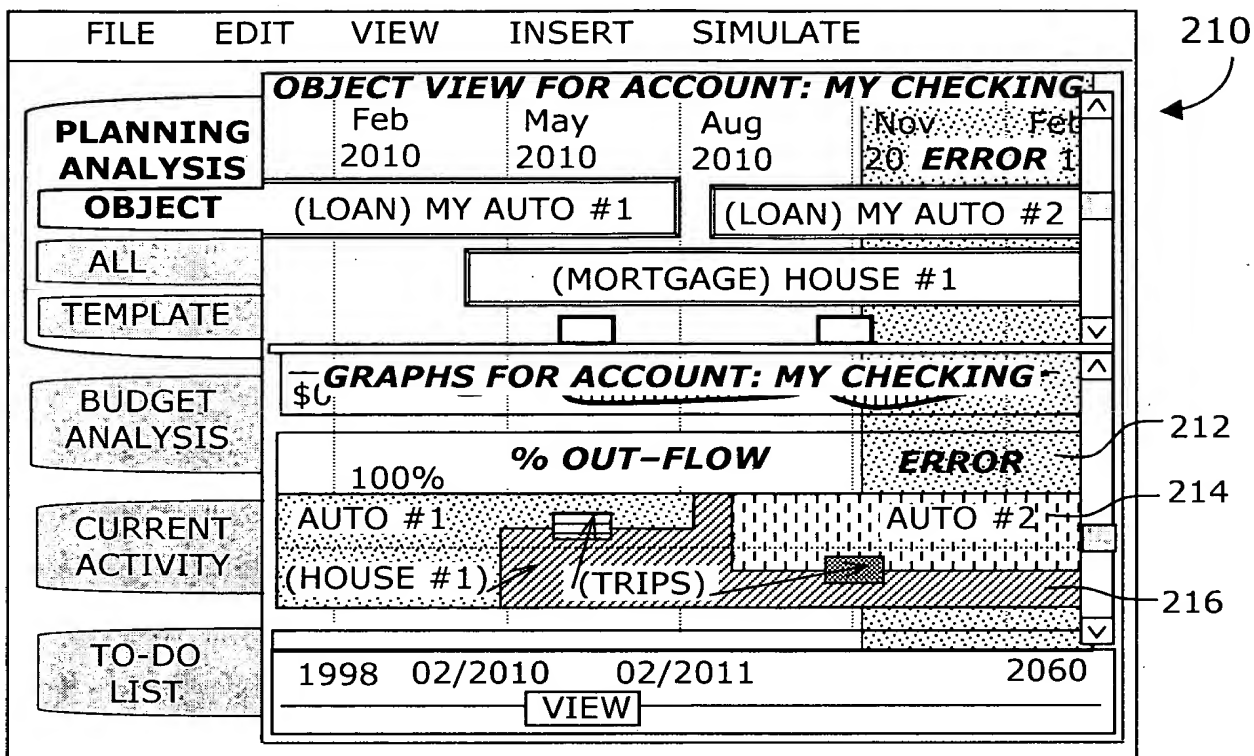


FIG. 21

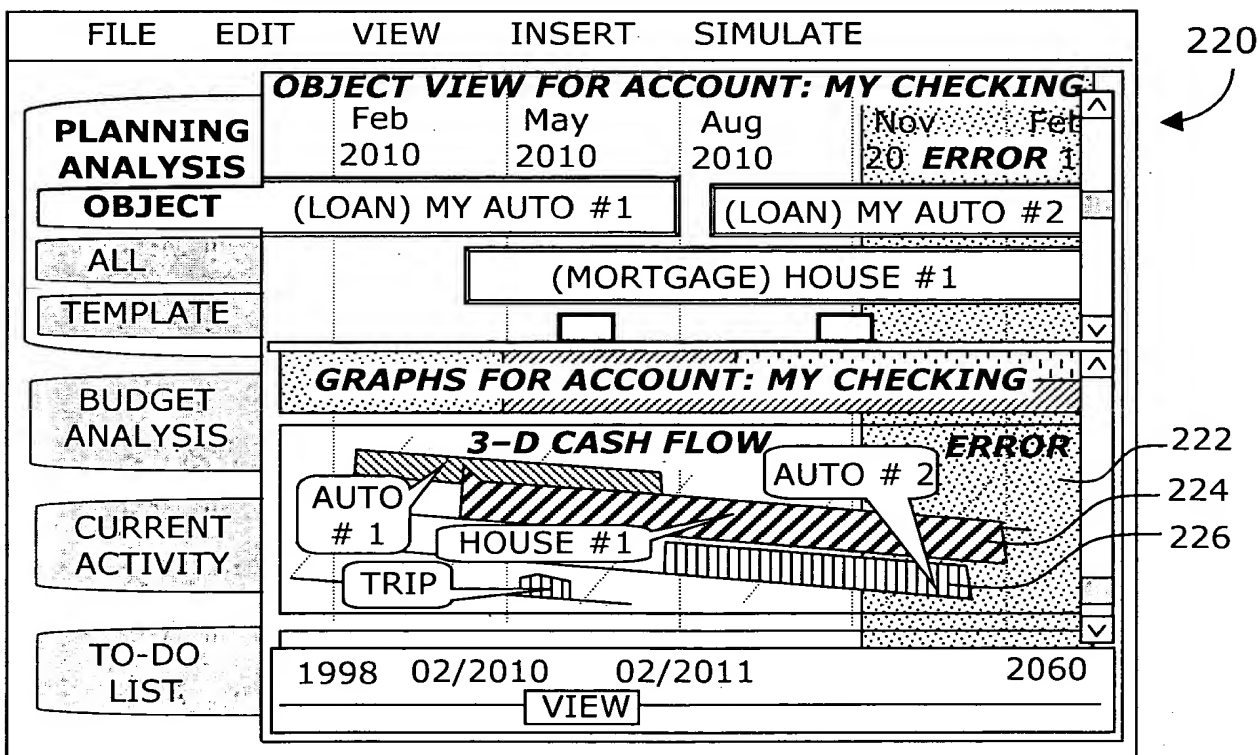


FIG. 22

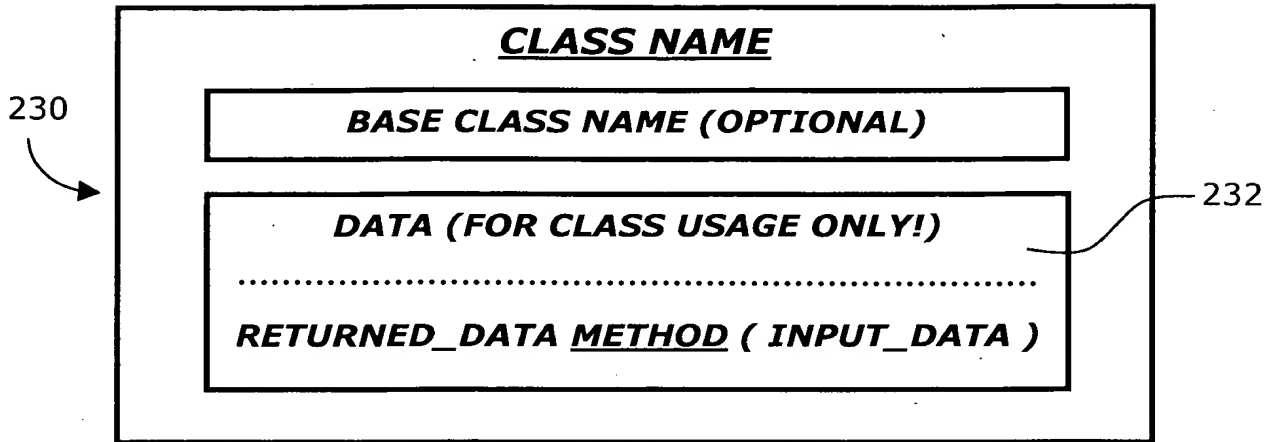


FIG. 23

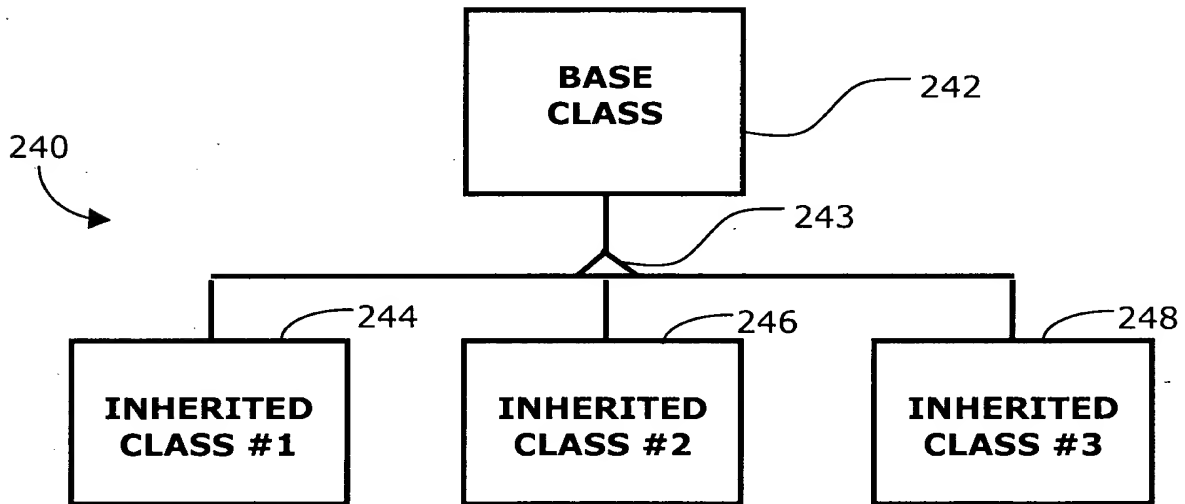


FIG. 24

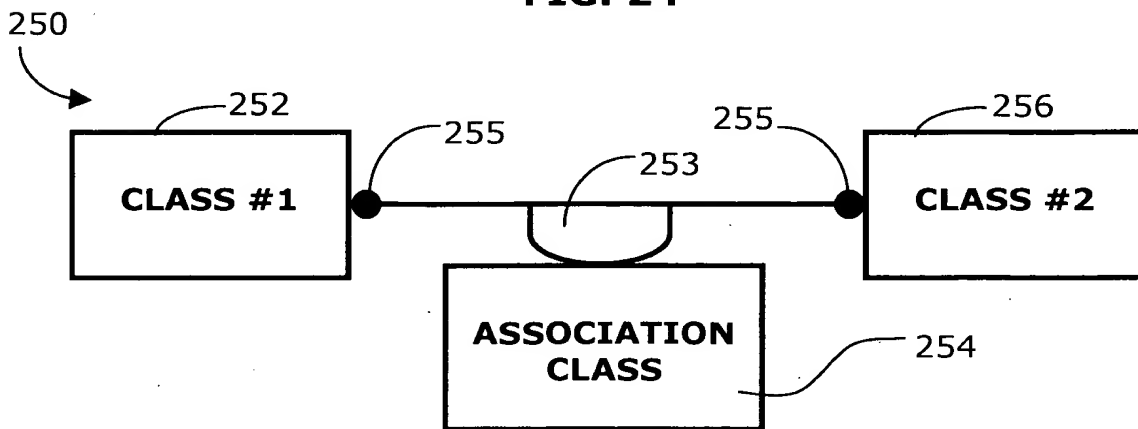


FIG. 25

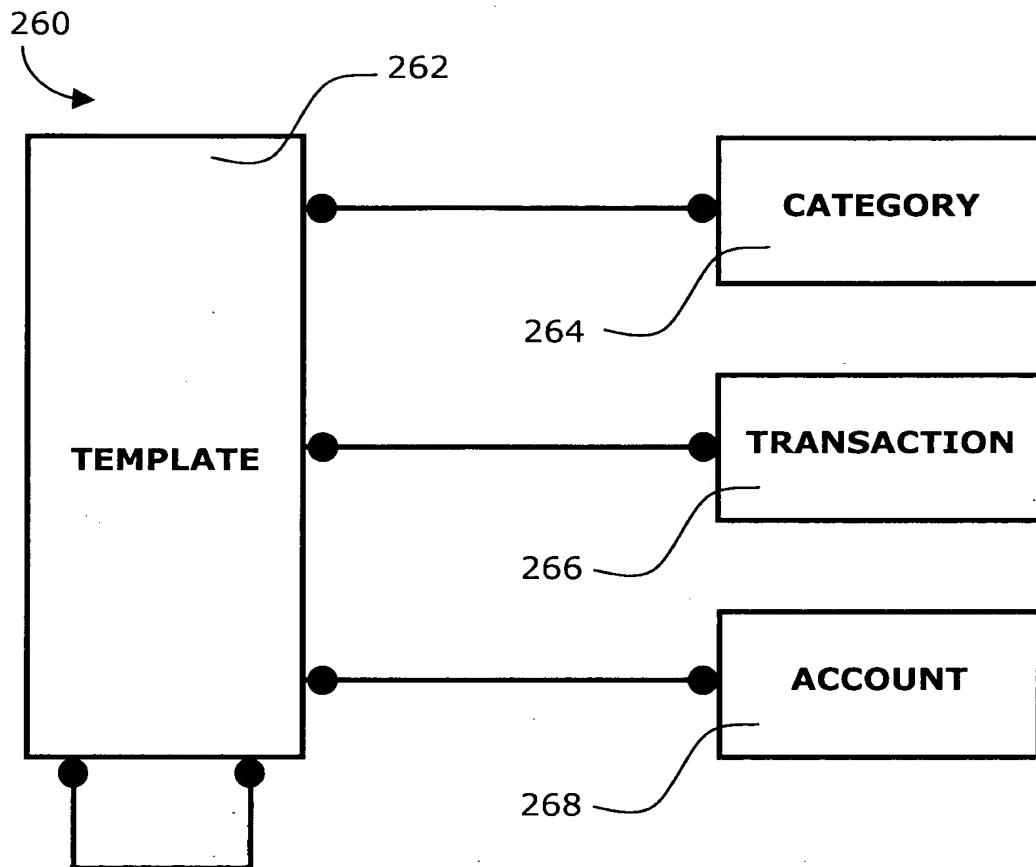


FIG. 26

15/28

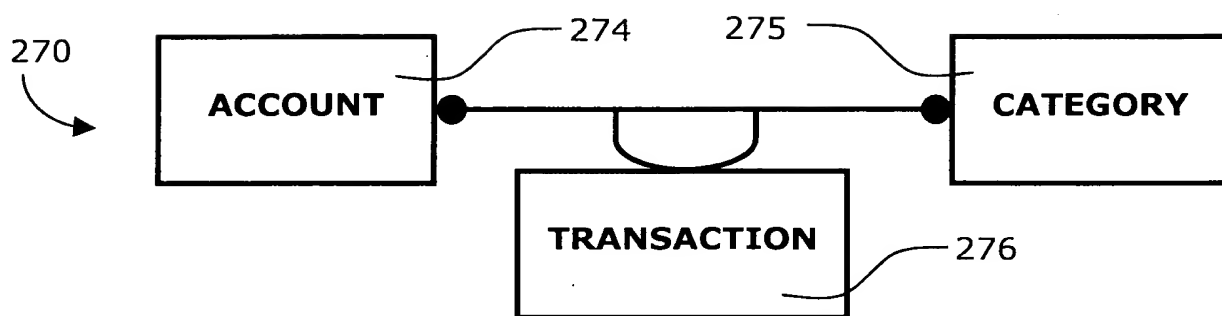


FIG. 27A

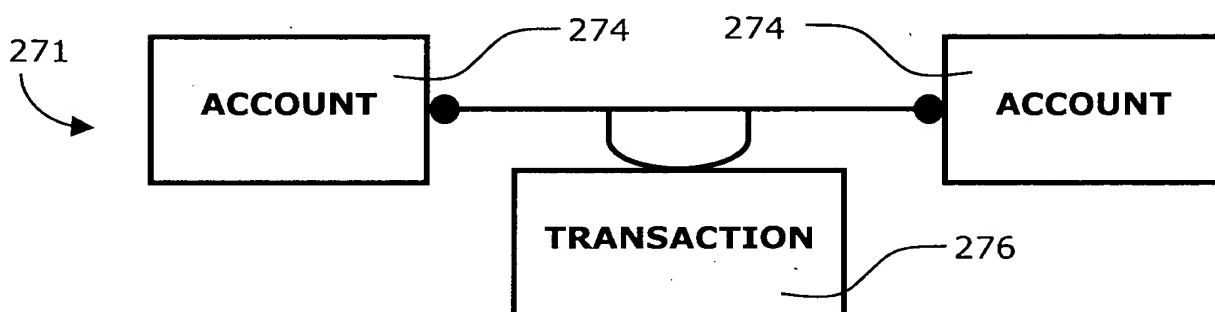


FIG. 27B

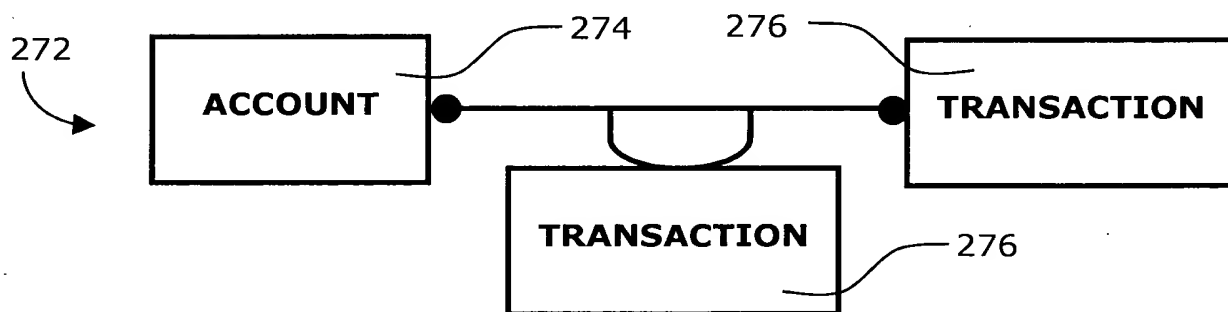


FIG. 27C

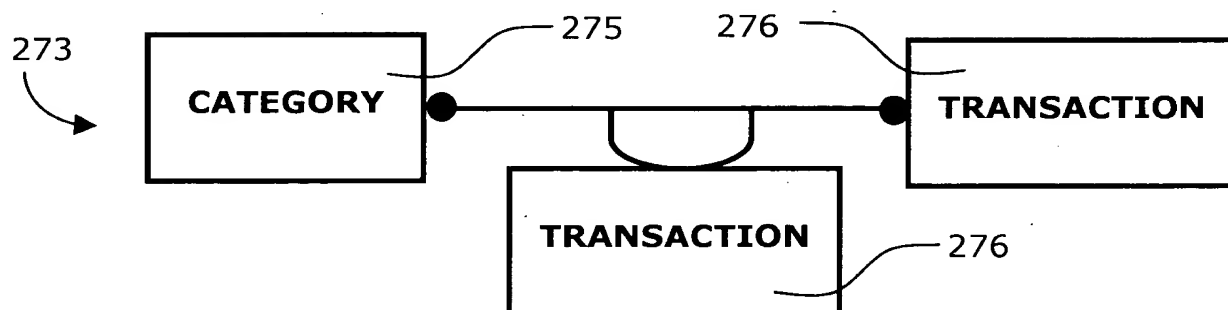


FIG. 27D

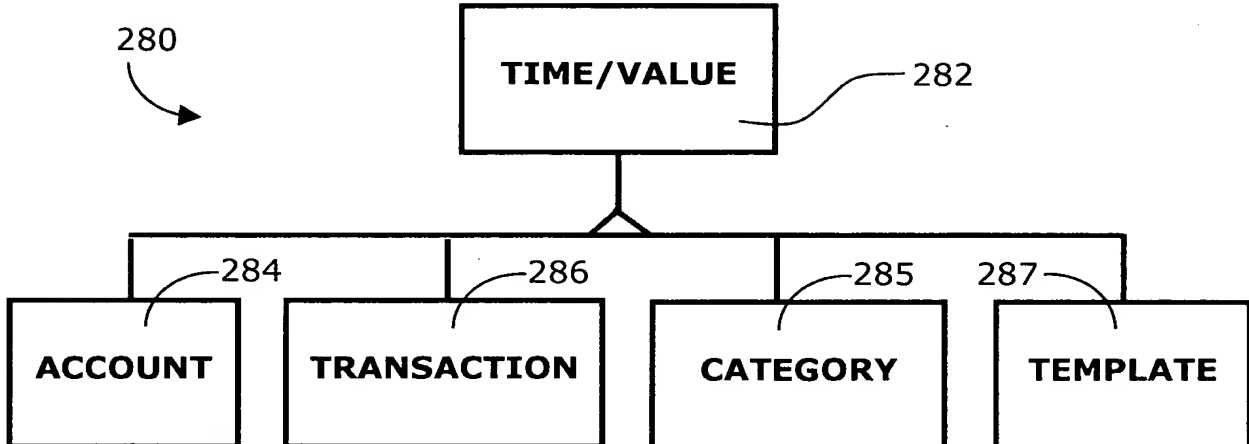


FIG. 28

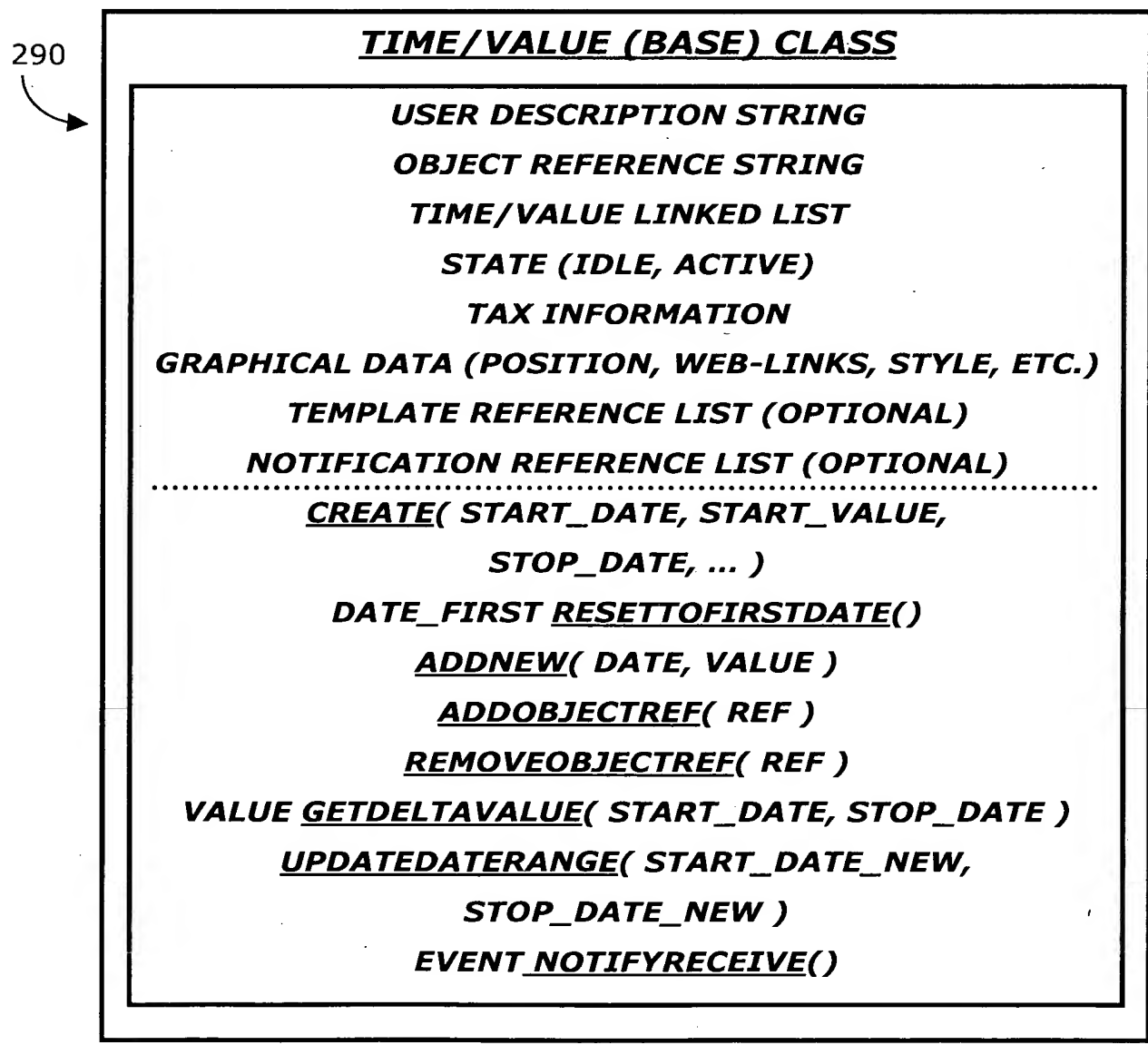


FIG. 29



300

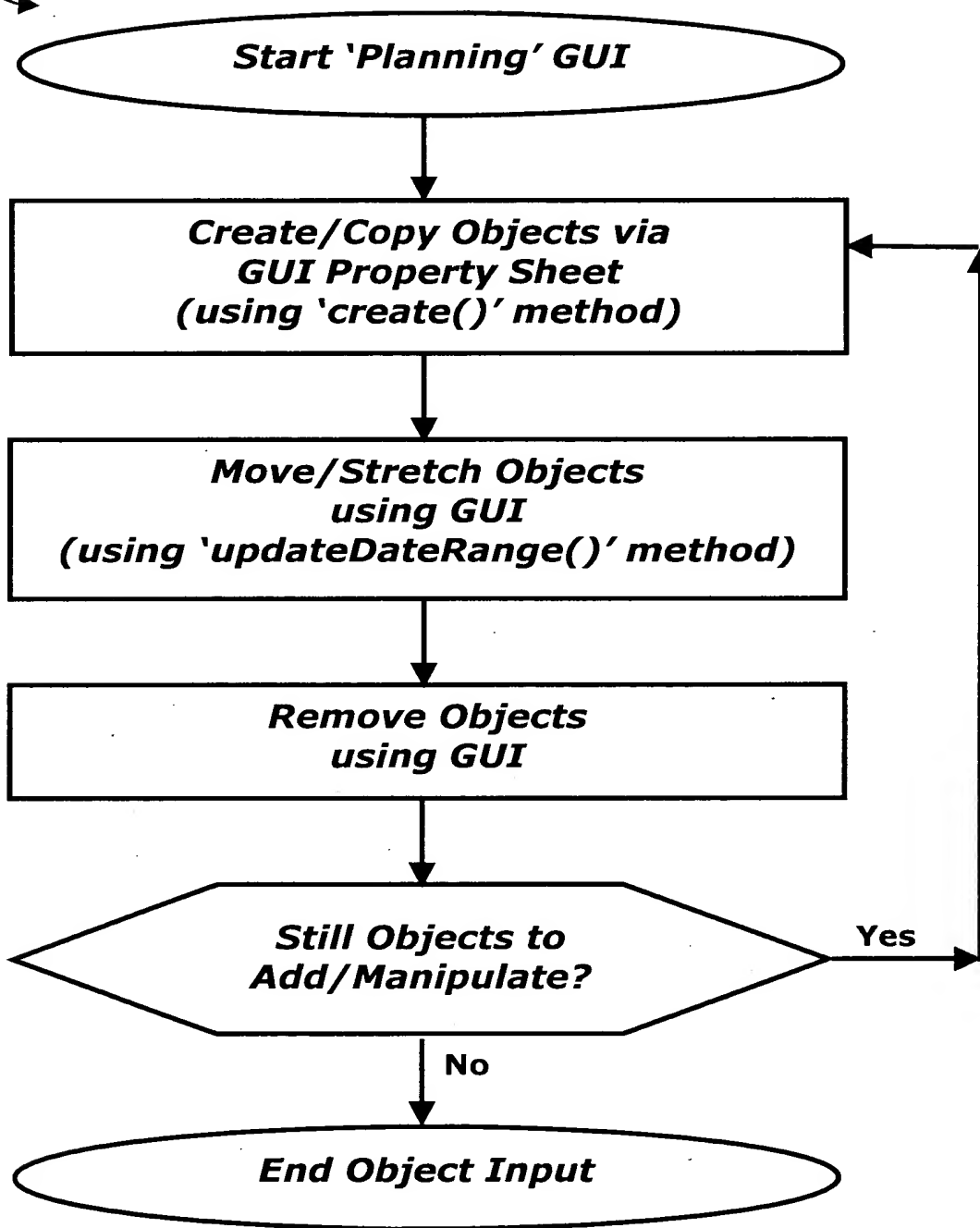


FIG. 30



18/28

310

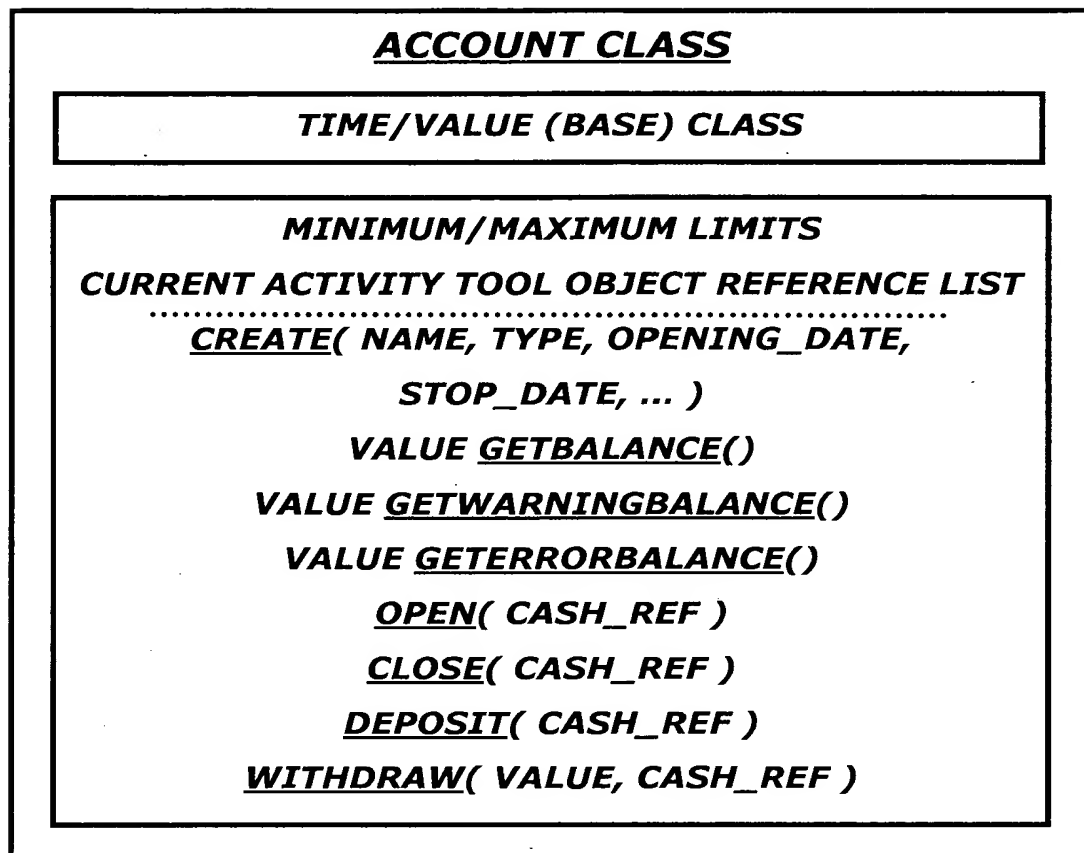


FIG. 31

320

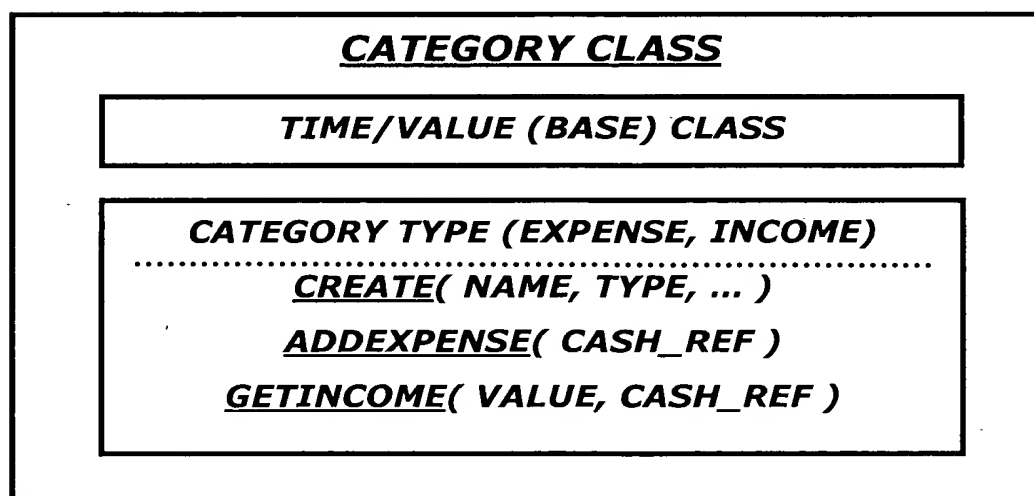


FIG. 32



330

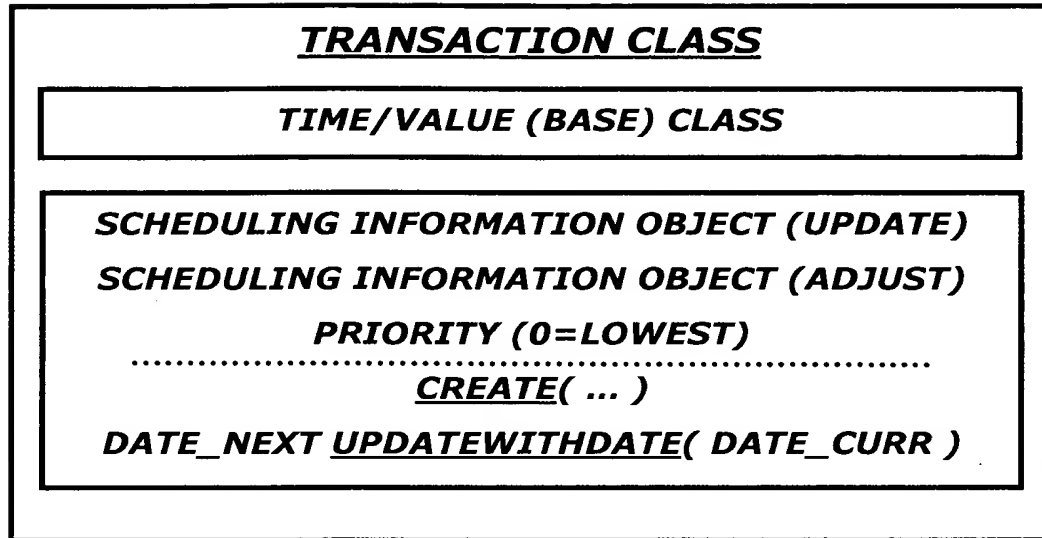


FIG. 33

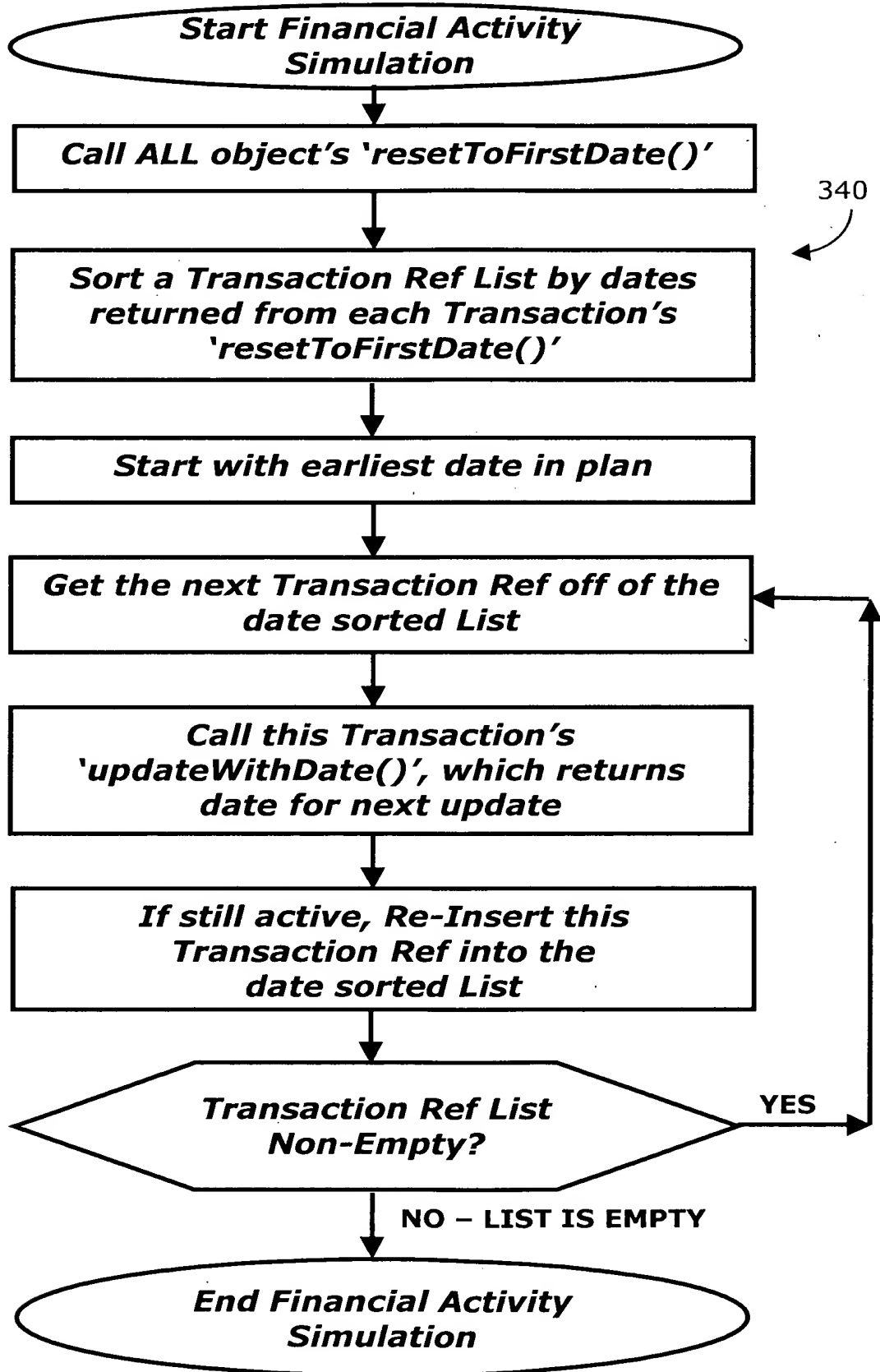


FIG. 34



21/28

350

### **SYSTEM INTERFACE CLASS**

**INFLATION-RATE-%/YEAR LINKED LIST**  
**MARKET-RETURN-%/YEAR LINKED LIST**  
**CURRENT AGE, RETIREMENT, LIFE EXPECTANCY**  
**'MISCELLANEOUS' CATEGORY REFERENCE**  
**REFERENCE CURRENCY (\$ OR FOREIGN)**  
.....  
**CREATE( ... )**  
**DATE GETCURRENTDATE()**  
**VALUE GETINFLATIONPCT( DATE )**  
**VALUE GETMARKETRETURNPCT( DATE )**  
**VALUE GETINFLATEDVALUE( VALUE\_FROM,**  
**DATE\_FROM, DATE\_TO )**  
**THROWWARNING( CODE )**  
**THROWERROR( CODE )**  
**PRINT( FORMAT\_STRING, ... )**  
**CREATECASH( VALUE, CASH\_REF )**  
**RETURNVALUE( VALUE, STRING\_REF )**  
**RETURNCASH( CASH\_REF, STRING\_REF )**  
**NOTIFYSEND( TARGET\_OBJECT\_REFERENCE, EVENT )**  
**NOTIFYALL( EVENT )**

FIG. 35

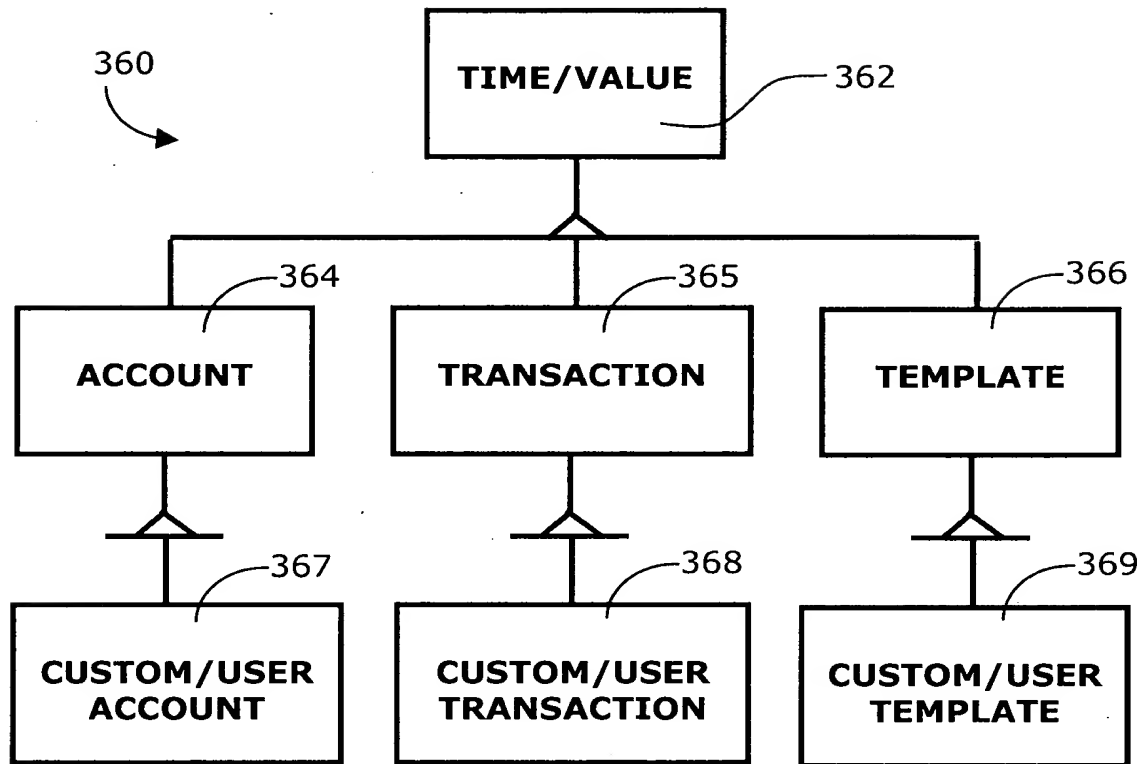


FIG. 36



370

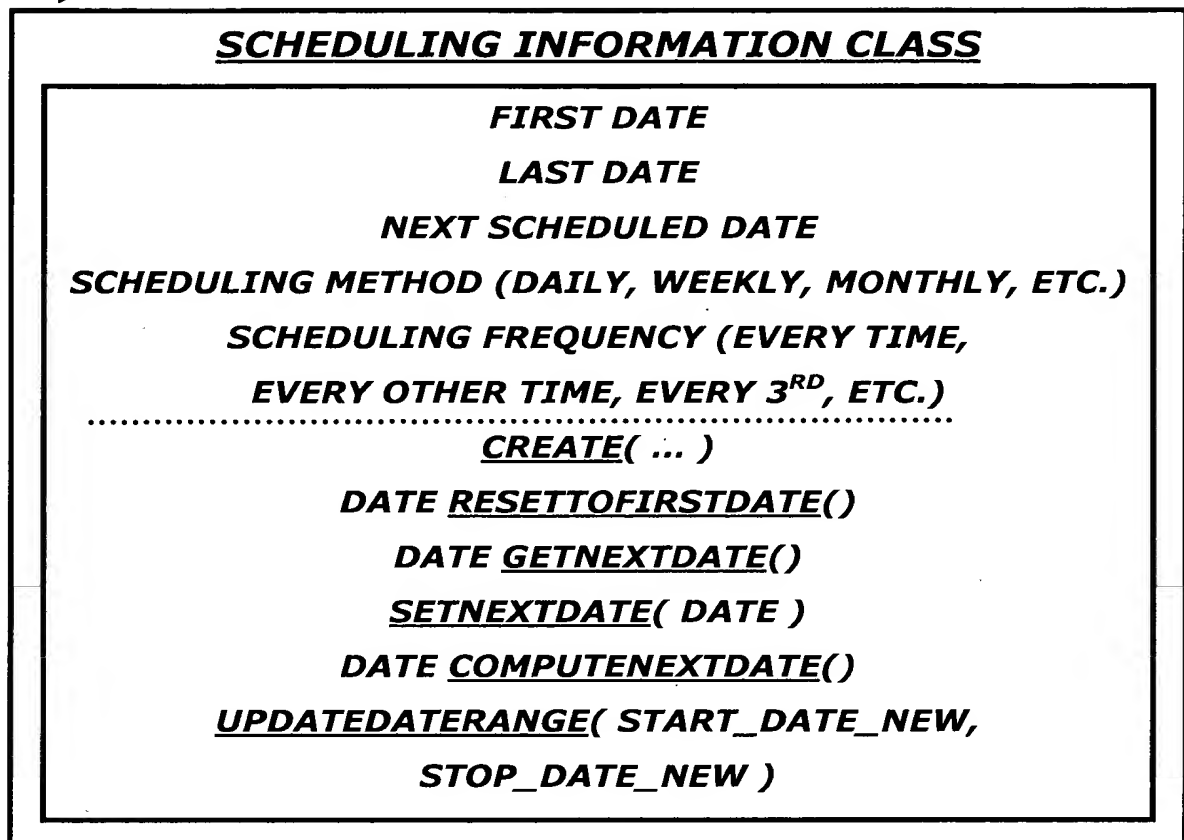


FIG. 37

380

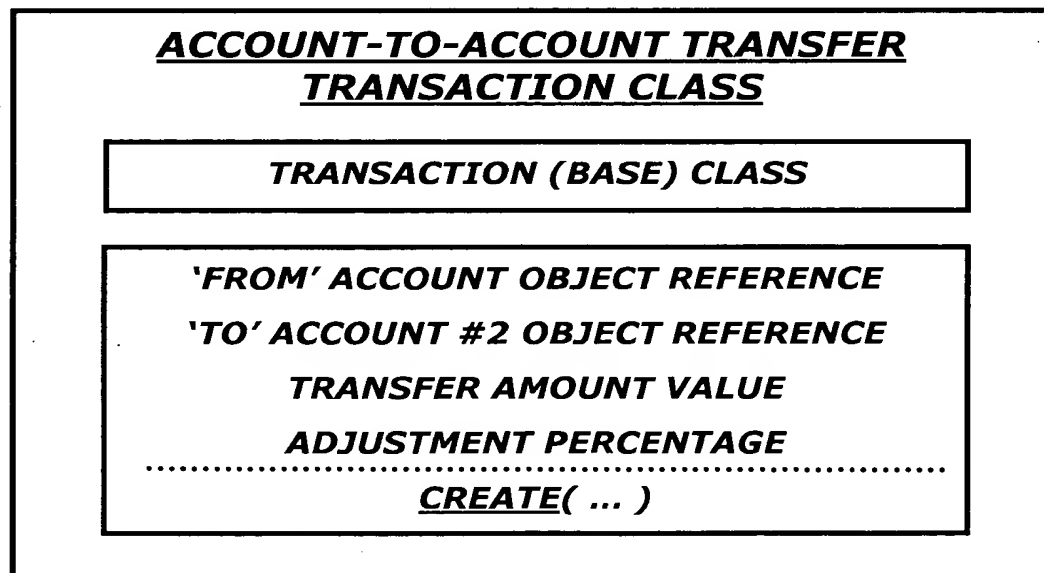


FIG. 38



24/28

390

FILE EDIT VIEW INSERT SIMULATE

ACCOUNT CATEGORY TEMPLATE TRANSACTION ACCT TO ACCT XFER

Aug Nov Feb

PLANNING ANALYSIS

ALL

OBJECT

**ACCOUNT TO ACCOUNT TRANSFER TRANSACTION:**

DESCRIPTION: MY CHECKING XFER TO MY SAVINGS

START DATE: 7/1/2006 END DATE: 7/30/2010

WITHDRAW FROM ACCOUNT: MY CHECKING

DEPOSIT TO ACCOUNT: MY SAVINGS

TRANSFER AMT (\$): (ENTER VALUE, OR HIT F1)

...EVERY: MONTH

ADJUST PCT (%): (ENTER VALUE, OR HIT F1)

...EVERY: YEAR

DONE

FIG. 39

400

FILE EDIT VIEW INSERT SIMULATE

**ACCOUNT TO ACCOUNT TRANSFER TRANSACTION:**

DESCRIPTION: MY CHECKING XFER TO MY SAVINGS

START DATE: 7/1/2006 END DATE: 7/30/2010

WITHDRAW FROM ACCOUNT: MY CHECKING

DEPOSIT TO ACCOUNT: MY SAVINGS

TRANSFER AMT (\$): (ENTER VALUE, OR HIT F1)

SELECT ONE OPTION FROM LIST FOR 'TRANSFER AMT (\$)' VALUE: R HIT F1

**SYSTEM METHODS**

ACCOUNT METHODS

TRANSACTION METHODS

RETURNED VALUES

getInflation()

**getInflatedValue()**

...

**getInflatedValue:**

Value From (\$) 150

Date From 1999

Date To (Current)

TO-DO LIST

1998 02/2010 02/

VIEW

402

FIG. 40



410

FILE EDIT VIEW INSERT SIMULATE

**PLANNING ANALYSIS**

Feb 2010 May 2010 Aug 2010 Nov 2010 Feb 2011

ALL

OBJECT: CATEGORY: CLOTHING

**ACCOUNT TO ACCOUNT TRANSFER TRANSACTION:**

DESCRIPTION: MY CHECKING XFER TO MY SAVINGS

START DATE: 7/1/2006 END DATE: 7/30/2010

WITHDRAW FROM ACCOUNT: MY CHECKING

DEPOSIT TO ACCOUNT: MY SAVINGS

TRANSFER AMT (\$): 150.00 (IN 1999 CASH)

...EVERY: MONTH

ADJUST PCT (%): (USER INFLATION RATE)

...EVERY: YEAR

DONE

FIG. 41

420

FILE EDIT VIEW INSERT SIMULATE

**PLANNING ANALYSIS**

Feb 2010 May 2010 Aug 2010 Nov 2010 Feb 2011

ALL

TEMPLATE

OBJECT: ACCOUNT(BANK): MY CHECKING

BUDGET ANALYSIS: MY CHECKING TO MY SAVINGS

CURRENT ACTIVITY: ACCOUNT(BANK): MY SAVINGS

TO-DO LIST

1998 02/2010 02/2011 2060

VIEW

FIG. 42

430



**Transaction Class**

```
class CTrans : public CtimeValue // A pure-virtual/abstract
                                // base class!
{
public:
    CTrans(
        const char *name, // Transaction's reference-name
        ... );             // More input parameters (not shown)
    virtual ~CTrans();

    virtual CDate updateWithDate( CDate date_curr ) = 0;
                                //PURE VIRTUAL!...Must inherit this class!

    virtual CDate resetToFirstDate();
                                // Inherited from Time/Value

    virtual void updateDateRange(
        CDate date_start, CDate date_stop );
                                // Inherited from Time/Value

protected:
    CScheduler m_schUpdate; // Schedules next update date
    CScheduler m_schAdjust; // Schedules next adjust date
    priority_t m_priority;   // Priority (0=lowest)

}; // END of 'CTrans' class
```

**FIG. 43**



27/28

**Account-to-Account Transfer  
Transaction Class**

```
class CTrans_acctToAcct : public CTrans
{
public:
    CTrans_acctToAcct(
        const char *name, // Transaction's reference-name
        ... );           // More input parameters (not shown)
    virtual ~CTrans_acctToAcct();

    virtual CDate updateWithDate( CDate date_curr );

    virtual CDate resetToFirstDate();
    virtual void updateDateRange(
        CDate date_start, CDate date_stop );

protected:
    CAccount *m_acctFrom; // Xfer 'From' this acct
    CAccount *m_acctTo;   // Xfer 'To' this acct
    value_t m_moneyToXfer; // Money to transfer at
                        // each update schedule
    value_t m_adjustPct;   // % to adjust xfer amount
}; // END of 'CTrans_acctToAcct' class
```

440

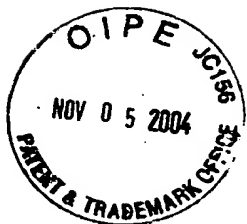
**FIG. 44**

**CASH CLASS**

450

```
AMOUNT VALUE
CURRENCY TYPE (DOLLARS, POUNDS, ETC.)
.....
CREATE()
ADDTOCASH( ADD_VALUE )
VALUE GETFROMCASH()
VALUE CURRENTAMOUNT()
```

**FIG. 45**



**Account-to-Account Transfer**  
**Transaction Class Method Example**

```
CDate CTrans_accntToAcnt::updateWithDate( CDate
date_curr ) {
    Cdate date_test =
        SYSINTF.getCurrentDate(); // Not used, just for demo
    if ( date_test == date_curr )
        SYSINTF.print( "Just a test...dates will be equal!" );
    // If simulated current date does NOT match our expected
    // current date, leave (an invalid condition)
    if ( date_curr != m_schUpdate.getNextDate() ) {
        SYSINTF.throwError( ERR_UNEXPECTED_DATE );
        return(date_curr); // Ends simulation for this transaction!
    }

    // If current simulation date matches or exceeds
    // our next adjustment date, adjust parameters
    if ( date_curr >= m_schAdjust.getNextDate() ) {
        m_moneyToXfer *= 1.0 + m_adjustPct / 100.0;
        m_schAdjust.computeNextDate(); } // Set the next
        // adjustment date

    // CREATE 'cash' data type (simulated cash amount = 0)
    CCash cash_xfer;
    // WITHDRAW cash FROM account (simulate cash > 0)
    m_acctFrom->withdraw( m_moneyToXfer, cash_xfer );
    // DEPOSIT cash TO account (simulated cash = 0)
    m_acctTo->deposit( cash_xfer );

    // LOG this transfer amount to the Time/Value (base) class
    addNew( date_curr, m_moneyToXfer );
    // Return the date that we wish the Cash-Flow Simulator to
    // call us with again
    return( m_schUpdate.computeNextDate() );
    // NOTE: Return from this method will cause 'cash_xfer'
    // to be destroyed, calling 'cash' class' destructor
    // method. A NON-ZERO simulated cash amount in
    // 'cash_xfer' would cause a system warning!
} // END of 'CTrans_accntToAcnt::updatePerDate()'
```